1.  How long did you spend on the code test?

5h 30m
30 minutes researching the problem
30 minutes experimenting with the graph.js API
2 hours coding the API backend
2 hours coding the Angular frontend
30 minutes code tidy and UI improvements

2.  What went well?

Using the ASP.NET Core starter template allowed me to make a SPA application with a backend using the inbuilt tooling a joy to work on. I could write the Angular code, change the UI and the app would reload itself when I saved the files.

Solving the problem, initially I wasn't sure if the solution should be based on a compound interest formula. I decided the problem was more about converting annual growth rates to monthly rates and applying this to the total investment as it accumulates over time.

The charts.js library worked well, it wasn't difficult to get it to read the server generated data and was very easy to customise the look and feel.

3.  Was there anything that was attempted but was not possible to get working in the time so is not visible in the code?

I attempted to display the end timescale using a plugin for charts.js. I think I could have got it working but decided I had already spent enough time on the main functionality. I also put in some unit tests to verify some calculations. I would usually write much more tests, including front end tests.

4.  What would you do to improve it / continue development?

Firstly, I would get a domain expert to verify the calculations and suitability for the product. I would then make sure I had plenty of test coverage. If required I would make the application ready for internationalisation as this is best put in place early.

The growth rates are hard coded into the application, it would probably be better if this were integrated with another service that provided updated growth rates, without the need to recompile and re-deploy my application.

There are certainly some performance tweaks to be made, and code refactoring to improve the maintainability. I had to put in some front-end hacks to clear the chart before it reloads, and an inelegant method of building the API request from the form to the API.